# REPORTE DE VULNERABILIDAD POR INYECCIÓN DE SQL

## INTRODUCCIÓN

En este reporte encontraremos todos los detalles relacionados a la vulnerabilidad por **INYECCIÓN DE SQL** en **DVWA.** Es importante destacar que esta prueba fue realizada en un ambiente controlado y especial para este tipo de laboratorios con intención totalmente ética de demostrar las **vulnerabilidades de este sistema**, **cuáles son las consecuencias** y **dar las recomendaciones necesarias para mejorar la protección** y evitar dichas vulnerabilidades.

## DESCRIPCIÓN DEL INCIDENTE

Durante las pruebas de seguridad que se realizaban en **DVWA**, se pudo evidenciar una vulnerabilidad de tipo **inyección de SQL**. Esta vulnerabilidad permite al atacante inyectar **"queries"** maliciosas a través del campo vacío donde uno coloca sus datos comprometiendo la **integridad** y **confidencialidad** de la base de datos.

## PROCESO DE REPRODUCCIÓN

Basicamente, para replicar y demostrar como se llego a dicha conclusión, insertamos el siguiente código en el campo denominado como USER ID

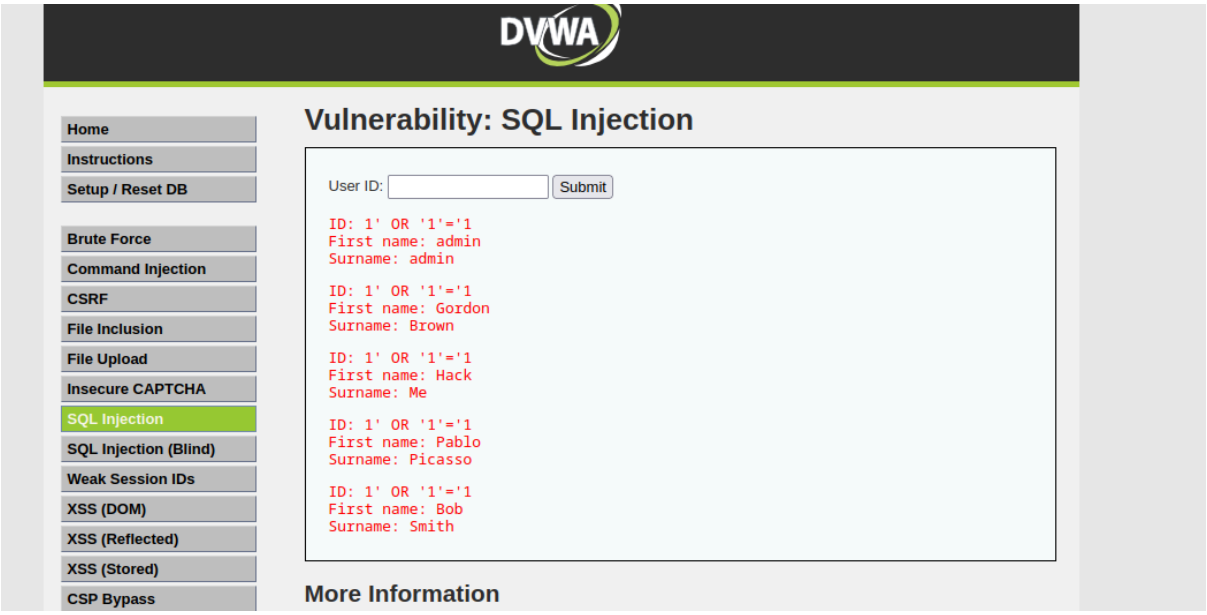
```
1' OR '1'='1
```

La inyección **SQL 1' OR '1'='1** es una técnica que consiste en manipular una consulta SQL introduciendo código malicioso en campos de entrada, como el campo USER ID. Al insertar esta cadena, la consulta resultante se convierte en algo como **SELECT * FROM users WHERE user_id = '1' OR '1'='1'** lo cual siempre será verdadero debido a la condición '1'='1'. Esto provoca que el sistema **devuelva todos los registros o permita el acceso sin validar correctamente el usuario**, representando una grave vulnerabilidad de seguridad. Para evitarlo, se deben utilizar consultas preparadas (prepared statements) y validar adecuadamente la entrada del usuario.

## IMPACTO DEL INCIDENTE

Explotar este tipo de vulnerabilidades, puede permitir al atacante:

- Permite el acceso no autorizado a datos sensibles.
- Puede alterar el comportamiento de dicha aplicación.
- Puede escalar y realizar acciones más graves como el borrado de datos, modificar la información o la extracción de esta.




## RECOMENDACIONES

Para prevenir ataques de inyección SQL, se recomienda implementar **consultas preparadas (prepared statements)** con parámetros, lo que evita que el código malicioso sea interpretado como parte de la consulta. Además, es fundamental **validar y sanitizar** todas las entradas del usuario, restringiendo los caracteres permitidos según el tipo de dato esperado. También se debe aplicar **el principio de menor privilegio** en la base de datos, otorgando a cada cuenta de usuario únicamente los permisos necesarios. Por último, es importante realizar **pruebas de seguridad periódicas** y utilizar herramientas de escaneo de vulnerabilidades para detectar posibles fallos en las aplicaciones.

## CONCLUSIÓN

En conclusión, esta práctica permite entender cómo funciona una inyección SQL y lo peligrosa que puede ser si no se controla adecuadamente la entrada del usuario. El ejemplo de **`1' OR '1'='1`** muestra lo fácil que es manipular una consulta para acceder a información sin autorización. Por eso es importante aplicar buenas prácticas como usar consultas preparadas y validar todo lo que el usuario ingresa. Hay que tener en cuenta que la seguridad no debe tomarse a la ligera, ya que un descuido en el código puede poner en riesgo toda la información del sistema.